Intel® 82806AA PCI 64 Hub (P64H)

Specification Update

July 2003

Notice: The Intel® 82806AA PCI 64 Hub may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are documented in this Specification Update.

Document Number: 290712-006

Contents

Revision History

Rev.	Draft/Changes	Date
-001	Initial Release	January 2001
-002	Removed the following as this information has been incorporated into the *Intel® 82806AA PCI 64 Hub (P64H)* datasheet (298025-002) —Specification Changes: Movement of APIC "DT" Bit Due to Microsoft* Windows NT*/ Windows* 2000 Overwrite —Documentation Changes: REQ64# Driven during Reset	March 2001
-003	Added: — Errata: P64H Blocks DO_SERR Message during Configuration Access to Non-Existing Device; P64H Deassert Message on Spurious Interrupt	May 2001
-004	Added: — Errata: P64H Incorrect End of Interrupt Status — Specification Clarification: P64H Arbitration Latency	July 2001
-005	Added: — Specification Clarification: P64H Bandwidth	April 2002
-006	Added: — Document Change: PCI Master Timeout	July 2003

Preface

This document is an update to the specifications contained in the documents listed in the following Affected Documents/Related Documents table. It is a compilation of device and document errata and specification clarifications and changes, and is intended for hardware system manufacturers and for software developers of applications, operating system, and tools.

Information types defined in the Nomenclature section of this document are consolidated into this update document and are no longer published in other documents. This document may also contain information that has not been previously published.

Affected Documents/Related Documents

Document Title	Document Number
Intel® 82806AA PCI 64 Hub (P64H) Datasheet	298025-002

Nomenclature

Specification Changes are modifications to the current published specifications. These changes will be incorporated in the next release of the specifications.

Errata are design defects or errors. Errata may cause the 82806AA P64H, behavior to deviate from published specifications. Hardware and software designed for used with any given stepping must assume that all errata documented for that stepping are present on all devices.

Specification Clarifications describe a specification in greater detail or further highlight a specification's impact to a complex design situation. These clarifications will be incorporated in the next release of the specifications.

Documentation Changes include typos, errors, or omissions from the current published specifications. These changes will be incorporated in the next release of the specifications.

Component Identification via Programming Interface

The 82806AA P64H may be identified by the following register contents:

Stepping	Vendor ID[1]	Device ID[2]	Revision Number[3]
B1	8086h	1360h (D31:F0) 1161h (D0:F0)	10h
B2	8086h	1360h (D31:F0) 1161h (D0:F0)	11h

NOTES:

1. The Vendor ID is located in the Vendor ID Register, address offset 00-01h in the PCI configuration space.
2. The Device ID is located in the Device ID Register, address offset 02-03h in the PCI configuration space.
3. The Revision Number is located in the Revision ID Register, address offset 08h in the PCI configuration space.

Component Marking Information

The 82806AA P64H may be identified by the following component markings:

Stepping	S-Spec	Top Marking	Notes
B1	SL3T5	FW82806AA, SL3T5	Production
B2	SL3VZ	FW82806AA, SL3VZ	Production

Summary Table of Changes

The following table indicates the Specification Changes, Errata, Specification Clarifications or Documentation Changes, which apply to the listed 82806AA P64H steppings. Intel intends to fix some of the errata in a future stepping of the component and to account for the other outstanding issues through documentation or Specification Changes as noted. This table uses the following notations:

Codes Used in Summary Table

Stepping

X:	Erratum, Specification Change or Clarification that applies to this stepping.
(No mark) or (Blank Box):	This erratum is fixed in listed stepping or specification change does not apply to listed stepping.

Status

Doc:	Document change or update that will be implemented.
Fix:	This erratum is intended to be fixed in a future stepping of the component.
Fixed:	This erratum has been previously fixed.
No Fix	There are no plans to fix this erratum.
Eval	Plans to fix this erratum are under evaluation.

Other

Shaded:	This item is either new or modified from the previous version of the document.

Number	SPECIFICATION CHANGES
	There are no specification changes in this Specification Update revision.

Number	Steppings			Status	ERRATA
	B1	B2			
1	X	X		Fixed	P64H I/O APIC Agent ID
2	X			Fixed	P64H Incorrect End of Interrupt Status
3		X		NoFix	P64H Repetitive Data
4		X		NoFix	P64H Blocks DO_SERR Message during Configuration Access to Non-Existing Device
5		X		NoFix	P64H Deassert Message on Spurious Interrupt

Number	SPECIFICATION CLARIFICATIONS
1	P64H Arbitration Latency
2	P64H Bandwidth

Number	DOCUMENTATION CHANGES
1	PCI Master Timeout

Specification Changes

There are no specification changes in this Specification Update revision.

This page is intentionally left blank.

Errata

1. Intel® P64H I/O APIC Agent ID

Problem: The P64H can incorrectly increment its I/O APIC agent ID when servicing interrupts under heavy I/O APIC traffic. This can cause the P64H to incorrectly drive an I/O APIC message when another agent (i.e., ICH) is performing an operation on the I/O APIC lines.

Implication: When this collision occurs the I/O APIC message, including the interrupt vector, can be corrupted. If the message is corrupted the processor will execute the incorrect interrupt service routine. The processor will also respond with an end of interrupt message with operation. This will lead to stuck level triggered interrupts and edge triggered interrupts that are not serviced.

Workaround: None identified.

Status: This erratum was fixed in the B-2 stepping.

2. P64H Incorrect End of Interrupt Status

Problem: A buffer full condition when receiving an End Of Interrupt (EOI) can occur in B1step P64H that results in the wrong EOI status being sent to the APIC and eventual hang of inbound PCI read requests.

Implication: PCI master will never be given the data for its read request and the system will hang with continuous retries on the PCI bus.

Workaround: None

Status: This erratum was fixed in B2 stepping.

3. Intel® P64H Repetitive Data

Problem: A rare buffer full condition can occur in P64H that results in a QWord of read completion data being repeated to a PCI master. This situation can happen if the following events occur:

- Processor write requests are unable to complete on the PCI bus (e.g., heavy traffic)
- Processor must issue five separate DWord or QWord PCI write requests
- The following must then occur in any order:
  - 5 processor DWord or QWord PCI write requests
  - 1 processor PCI read request
  - 1 APIC Buffer Flush Acknowledge (result of a PCI interrupt)
- P64H then receives a 3-QWord read completion resulting from a PCI master read request to an address which is offset 3 QWords below a 64 byte cache line boundary (PCI address [5:3] = 101).

Implication: The third QWord of the PCI read completion will be repeated and delivered to the PCI master as the fourth QWord.

Workaround: Disable P64H I/O APIC

Status: There are currently no plans to fix this erratum

4. P64H Blocks DO_SERR Message during Configuration Access to Non-Existing Device

Problem: If a Configuration Read to a non-existing device occurs concurrently with a write data parity error, and if the P64H is configured to generate SERR# on those writes, it is possible that SERR# may not be generated and all subsequent SERR#s will be blocked until the SSE bit (Signal System Error, bit 14) in the P64H Secondary Status Register (offset 1E-1Fh) is cleared.

Implication: NMI generation could be blocked until the SSE bit is cleared.

Workaround: None

Status: There are no plans to fix this erratum.

5. P64H Deassert Message on Spurious Interrupt

Problem: If an interrupt is asserted, then deasserted before it can be serviced (spurious interrupt), the P64H may incorrectly send an interrupt deassert message rather than an interrupt assert message.

Implication: Possible system hang due to a spurious interrupt

Workaround: Disable I/OAPIC and route interrupts via ICH.

Status: There are no plans to fix this erratum.

Specification Clarifications

1. P64H Arbitration Latency

The P64H prioritizes incoming transactions from PCI (both read and write) over CPU initiated transactions. As such, PCI devices may get much higher throughput. Additionally, the PCI posted buffer to main memory and the read return buffer for CPU initiated reads are shared.

Under certain conditions, as PCI devices fill this buffer, CPU read transactions may be significantly delayed. This typically only happens for CPU-initiated-transactions which result in delayed transactions, as the first transactions will be retried by the target. It is the subsequent re-initiation of the transaction that may be significantly delayed.

For systems which incorporate devices behind the P64H that may issue delayed transactions, (such as PCI-PCI bridges) care should be taken with these devices to not time out the CPU initiated transaction or generate any system faults (such as SERR#).

2. P64H Bandwidth

P64H sustained PCI bandwidth is ~300 MB/s for reads or writes. The actual bandwidth will be system configuration dependent. For those configurations that show less bandwidth, improvement may be possible by adjusting the following P64H register bits:

CNF – P64H Configuration Register (D31:F0), Address Offset 50-51h, Bit [2], Delay Transaction (DT) Depth. Change bit 2 from "0" to "1".

Soft_DT_Timer – Soft Delayed Transaction Timer Register (D31:F0), Address Offset 80h, bits [1:0], 1 bit Soft Delayed Transaction Timer. Change bits [1:0] from default to a lesser value.

Changing these registers as recommended can increase overall bandwidth by reducing the time slice specified for Delayed Transactions.

This page is intentionally left blank.

Documentation Changes

1. PCI Master Timeout

Replace the Bridge Control Register (D31:F0), address offset: 3E-3Fh bit 9 in Section 2.2.8 in the Intel® 82806AA PCI 64 Hub (P64H) Datasheet with the following:

9	**PCI Master Timeout— R/W:** This bit sets the maximum number of PCI clock cycles that the P64H will wait for the initiator to repeat a delayed transaction request. The counter starts when the delayed transaction completion is at the head of the queue. If the master does not repeat the transaction at least once before the counter expires, the P64H discards the transaction from its queues. 0= PCI master time out value is between 2^{15} and 2^{16} PCI clocks. 1= PCI master time out value is between 2^{10} and $2^{11.}$